

Data Sheet

BUSINESS CHALLENGE

There is an increasing demand from users of modern business IT applications for easier to use applications which integrate with other business systems. As a result IT organizations are asked to deliver modern user interfaces and integrate multiple business systems.

At the same time IT continually strives to reduce operational costs and risks by standardizing on common platforms for all their applications – allowing the same tools and processes to be used regardless of the programming language used to build the applications. Microsoft's .NET framework and the Java Virtual Machine (JVM) are widely seen as the standard frameworks of choice.

Often the applications run on aging or even unsupported hardware and software environments. Moving to commodity hardware and standard frameworks can return considerable cost savings and business agility.

Previously, IT organizations and ISVs considered that their only option to deliver the new features and deploy into the standard frameworks was to rewrite business applications in Java or C#. This introduces considerable cost and risk for little added value. With Visual COBOL the application can remain in COBOL and the application provider can choose to deploy in native or managed code for .NET or JVM platforms. This way the benefits of the platform, as well as the traditional strength and reliability of the existing application, are retained.

Skilled IT programming resources in any enterprise business language are becoming increasingly hard to find. By standardizing on the leading integrated development environments (IDEs) – Visual Studio and Eclipse – and ensuring COBOL is a first-class citizen in those IDEs, IT teams and ISVs can easily move programmers between projects, regardless of the programming languages used. This increases business agility and reduces costs.

PRODUCT OVERVIEW

Previously the development tools needed to build, test and deploy business applications on UNIX or Linux platforms as native or managed code to run on a JVM, have been very limited. Eclipse represents the best cross-language and cross-platform IDE available today. Micro Focus Visual COBOL's integration in Eclipse delivers the richest development experience for COBOL programmers targeting deployment of applications on Linux or UNIX, or deploying as a JVM application. In parallel, the COBOL language has been enhanced to make it even easier to use for JVM applications.

BUSINESS BENEFIT

Visual COBOL provides familiar, high-performance developer tools to address the needs of modern business IT. By combining rich tools with a modern language existing applications can be reused in new ways, integrated with other systems and deployed to cost-effective and robust execution environments.

Visual COBOL for Eclipse is a part of the Visual COBOL product portfolio from Micro Focus which includes testing and developer productivity tools.

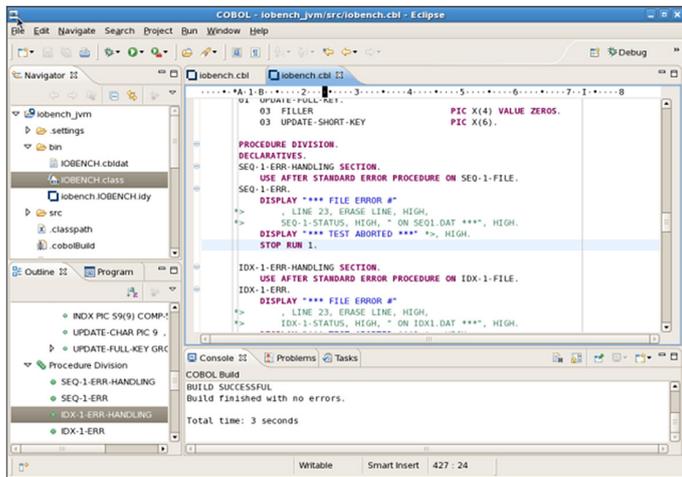
FEATURE OVERVIEW

- Fully integrated COBOL development environment delivers high programmer productivity by exploiting Eclipse tools, including providing instant feedback of syntax errors while editing
- Enhanced COBOL syntax for easier development of Object Oriented (OO) COBOL or to integrate OO and Java classes into traditional COBOL applications
- Visual COBOL Development Hub supports high performance development of UNIX or Linux server applications from Windows or Linux desktop clients (add-on option)
- Build modern, highly integrated mixed COBOL and Java applications with COBOL for JVM.

Data Sheet

DETAILED FEATURE OVERVIEW

Fig 1: Modern COBOL IDE



COBOL IDE

The Eclipse editor has been extended to enhance COBOL programmer productivity. Changes include visual indication of COBOL margins which are sensitive to the COBOL margin directive in current use, enablement of the Eclipse program navigation features for COBOL, COBOL comment structures, column block marking, and more.

Background parsing continuously ensures that the code compiles cleanly. Tools such as the Outline view, Find References, and in place copy file expansion are tuned for COBOL productivity.

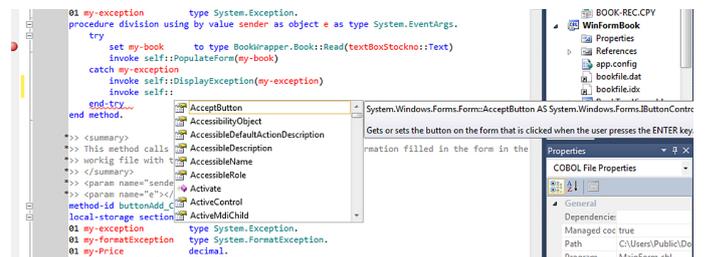
The Eclipse debugger fully supports COBOL in data queries, watch windows, breakpoints, watchpoints and more, as well as having control over the program execution flow both for native code and COBOL JVM.

MODERN COBOL

COBOL has historically been case-insensitive, causing interoperation with .NET or JVM classes and methods to be less intuitive than it should be. For example, method or member names needed to be enclosed within quotation marks. Visual COBOL removes these restrictions so the code is more natural for JVM and .NET programming, while keeping the tradition of making it easy to understand.

Unnecessary COBOL elements such as 'REPOSITORY' are now optional which greatly reduces the size and complexity of a managed COBOL program, improves readability, and simplifies the learning process for Java, C# or Visual Basic programmers.

Fig 2: Modern OO COBOL with IntelliSense



COBOL FOR JVM DEVELOPMENT

Visual COBOL further extends deployment flexibility by offering deployment of COBOL applications within the JVM. COBOL applications are compiled directly to Java byte code and can be reused by Java classes within desktop applications, or within web applications hosted under a Java Application Server.

DISTRIBUTED DEVELOPMENT

For applications being deployed on Linux or UNIX servers, it is important to be able to develop those applications on the same operating platform which is used in the production environment. This avoids the need to reproduce databases, middleware and other systems support on the developer's local computer. It implements complex processes for migrating, building and testing source code as it moves from the programmer's machine into system testing or integration testing.

Data Sheet

The [Visual COBOL Development Hub \(DevHub\)](#), available as an add-on, allows the project source code, databases and other services to stay on the target platform while the Eclipse environment runs on the programmer's Windows or Linux desktop for maximum IDE performance.

RELATIONAL DATABASE SUPPORT

The OpenESQL toolset provides support for COBOL applications accessing any ODBC or enabled relational database using the familiar 'EXEC SQL' syntax.

EXTEND® COMPATIBILITY

Many *extend* applications could already be moved to Visual COBOL, but where the applications used some of the advanced features, updates to the source code were needed. Now, the compatibility of *extend* with Visual COBOL data types and selected runtime calls means that less work is required to make the move. Compiler warnings highlight code which can be used in Visual COBOL.

The inclusion of Vision, the *extend* file handler, allows existing data files to be reused with Visual COBOL without requiring any risk or effort to migrate the data files.

RM/COBOL COMPATIBILITY

Visual COBOL introduces new RM/COBOL compatibility, including syntax extensions such as XML Extensions and CALL runtime behavior, making it easier to move existing code to Visual COBOL. The RM file handler removes the need to migrate the application's data.

COBOL SERVER

Visual COBOL is built on a new Micro Focus COBOL platform known as COBOL Server. This major reworking of the COBOL platform simplifies installation and reduces memory footprint.

A standalone COBOL Server Runtime is available for deploying applications developed with Visual COBOL.

PLATFORMS

Eclipse Client

- Windows Vista
- Windows XP
- Windows 7
- Windows 8.1
- Windows Server 2003
- Windows Server 2008
- Windows Server 2012 R12
- SUSE 11 SP2/SP3
- Red Hat 5.10, 6.5

Development Hub

- AIX 6.1/7.1
- SUSE 11 SP2/SP3 on Intel
- Redhat 5.10, 6.5 on Intel
- Oracle Linux 6.5 on Intel
- SUSE 11 SP2/SP3 on OS/390
- Red Hat 6.5 on OS/390
- Solaris 10, 11.1 SPARC
- Solaris 11.1 on Intel
- HP/UX 11i v3 (11.31)

For additional information please visit: www.microfocus.com

© 2014 Micro Focus Limited. All rights reserved. MICRO FOCUS, the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus Limited or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. All other marks are the property of their respective owners.